CLAIMS

What is claimed is:

1) In a multi-processor server system, a method for processing a connection comprising:

a) receiving a connection indicator packet on a listener of said server system wherein said connection indicator packet is associated with a connection to be established;

b) in response to said connection indicator, generating a connection data structure for said connection to be established and assigning said connection to be established to a processor of said multi processor server;

c) said connection to be established transmitting a first acknowledgement packet in response to said connection indicator wherein said listener is free to process subsequent packets;

d) said connection to be established receiving a second acknowledgement packet to establish a connection path on said processor; and

e) said connection path getting established with processing restricted to the said connection path and said connection path subsequently processing a plurality of data packets on said processor associated with said connection to be established, wherein said data packets are identified with said connection path based on said data structure.

2) The method as described in Claim 1 wherein said connection path is a perimeter for processing data packets associated with said connection to be established.

3) The method as described in Claim 2 wherein said perimeter is a defined set of modules that each packet of said connection is processed.

4) The method as described in Claim 1 wherein said connection indicator is a SYN packet.

5) The method as described in Claim 4 wherein said first acknowledgement packet is a SYN_ACK packet.

6) The method as described in Claim 5 wherein said second acknowledgement packet is a SYN_ACK_ACK packet.

7) The method as described in Claim 1 wherein said listener comprises a socket, TCP, or IP control block.

8) The method as described in Claim 1 wherein said listener is not single threaded while processing said connection to be established.

9) The method as described in Claim 1 wherein said data structure is a conn_t.

10) A computer system comprising a plurality of processors coupled to a bus and a memory coupled to a bus and comprising instructions that when executed, implement a method of connecting a multi-processor server to a client comprising:

a) receiving a connection indicator packet on a listener of said server system wherein said connection indicator packet is associated with a connection to be established;

b) in response to said connection indicator, generating a connection data structure defining a connection path for said connection and assigning said path to a processor of said multi processor server;

c) said path transmitting a first acknowledgement packet in response to said connection indicator wherein said listener is free to process subsequent packets;

d) said path receiving a second acknowledgement packet to establish said connection on said processor; and

e) said path processing a plurality of data packets on said processor associated with said connection, wherein said data packets are identified with said path based on said data structure.

11) The computer system as described in Claim 10 wherein said path is a perimeter for processing data packets associated with said connection.

12) The computer system as described in Claim 11 wherein said perimeter is a defined set of modules that each packet of said connection is processed.

13) The computer system as described in Claim 10 wherein said connection indicator is a SYN packet.

14) The computer system as described in Claim 13 wherein said first acknowledgement packet is a SYN_ACK packet.

15) The computer system as described in Claim 14 wherein said second acknowledgement packet is a SYN_ACK_ACK packet.

16) The computer system as described in Claim 10 wherein said listener comprises a socket, TCP, or IP control block.

17) The computer system as described in Claim 10 wherein said listener is not single threaded while processing said connection.

18) The computer system as described in Claim 10 wherein said data structure is a T_CONN data structure.

19 A method for reducing load on an acceptor of a multi-processor server system comprising:

a) receiving a connection indicator packet on said listener of said server system wherein said connection indicator packet is associated with a connection to be established;

b) in response to said connection indicator, generating a connection data structure defining a connection path for said connection and assigning said path to a processor of said multi processor server;

c) said path transmitting a first acknowledgement packet in response to said connection indicator wherein said listener is free to process subsequent packets;

d) said path receiving a second acknowledgement packet to establish said connection on said processor; and

e) said path processing a plurality of data packets on said processor associated with said connection, wherein said data packets are identified with said path based on said data structure.

20) The method as described in Claim 19 wherein said path is a perimeter for processing data packets associated with said connection.

21) The method as described in Claim 20 wherein said perimeter is a defined set of modules that each packet of said connection is processed.

22) The method as described in Claim 19 wherein said connection indicator is a SYN packet.

23) The method as described in Claim 22 wherein said first acknowledgement packet is a SYN_ACK packet.

24) The method as described in Claim 23 wherein said second acknowledgement packet is a SYN_ACK_ACK packet.

25) The method as described in Claim 19 wherein said listener comprises a socket, TCP, or IP control block.

26) The method as described in Claim 19 wherein said listener is not single threaded while processing said connection.

27) The method as described in Claim 19 wherein said data structure is a T_CONN data structure.